EECE256 Midterm; Section 101 and 102

Dr. Sidney Fels & Steve Oldridge

90 min

closed book, calculator allowed

Student Name:_____ **S#:**_____ **Section #**____

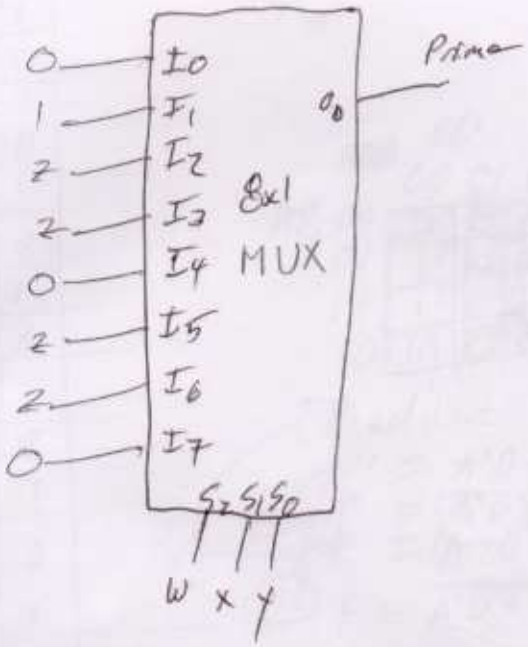
Question	Mark
1 (12)	
2 (8)	
3 (4)	
4 (12)	
Total (36)	

1. You are required by your manager to design a **4-bit prime number** checker (i.e. you want to output 1 if the binary number is prime; don't forget, 0 and 1 are not prime numbers).

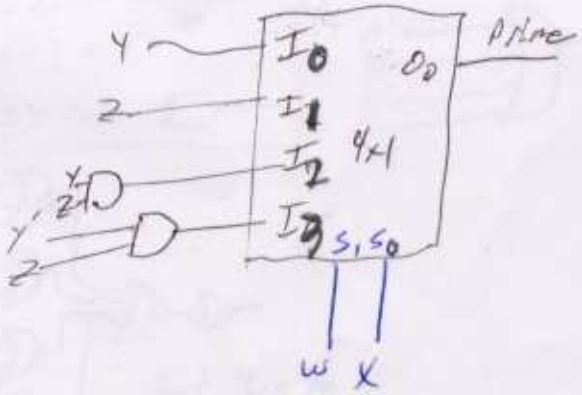
   a. Show the truth table for the prime number checker. (4 marks)

w x y z	Prime	I_1^8	I_1^4	
0 0 0 0	0	0		
0 0 0 1	0			
0 0 1 0	1	1		y
0 0 1 1	1			
0 1 0 0	0	z		
0 1 0 1	1			z
0 1 1 0	0	z		
0 1 1 1	1			
1 0 0 0	0	0		
1 0 0 1	0			y·z
1 0 1 0	0	z		
1 0 1 1	1			
1 1 0 0	0	z		
1 1 0 1	1			y'·z
1 1 1 0	0	0		
1 1 1 1	0			

b. Design the circuit using a **single 8x1 multiplexer** and a minimal number of extra **AND, OR** or **NOT** gates if needed (i.e., no NAND, NOR, XOR or XNOR, etc.). Show your work including any simplifications done. You must show the final logic diagram with all the pins of the multiplexer labelled properly. State any assumptions that you make (4 marks)



c. Design the circuit using a **single 4x1 multiplexer** and a minimal number of extra AND, OR or NOT gates if needed (i.e., no NAND, NOR, XOR or XNOR, etc.). Show your work including any simplifications done. You must show the final logic diagram with all the pins of the multiplexer labelled properly. State any assumptions that you make (4 marks)



Design a two level circuit

2. Draw a logic diagram using a **minimal** number of NOR gates (2, 3 and 4 input gates are allowed) to implement the following function (8 marks):

$$F(A,B,C,D) = [((A' + B' + C) \cdot D) \oplus (A \cdot B)'] + (C' \cdot D')$$

T_1 T_2 T_3 T_4 T_5

A B C D	T_1	T_2	T_3	T_4	T_5	F
0 0 0 0	1	0	1	1	1	1
0 0 0 1	1	1	1	0	0	0
0 0 1 0	1	0	1	1	0	1
0 0 1 1	1	1	1	0	0	0
0 1 0 0	1	0	1	1	1	1
0 1 0 1	1	1	1	0	0	0
0 1 1 0	1	0	1	1	0	1
0 1 1 1	1	1	1	0	0	0
1 0 0 0	1	0	1	1	1	1
1 0 0 1	1	1	1	0	0	0
1 0 1 0	1	0	1	1	0	1
1 0 1 1	1	1	1	0	0	0
1 1 0 0	0	0	0	0	1	1
1 1 0 1	0	0	0	0	0	0
1 1 1 0	1	0	0	0	0	0
1 1 1 1	1	1	0	1	0	1

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	1	0
10	1	0	0	1

(Product of sums)

$F' = A'D + B'D + C'D + ABCD'$

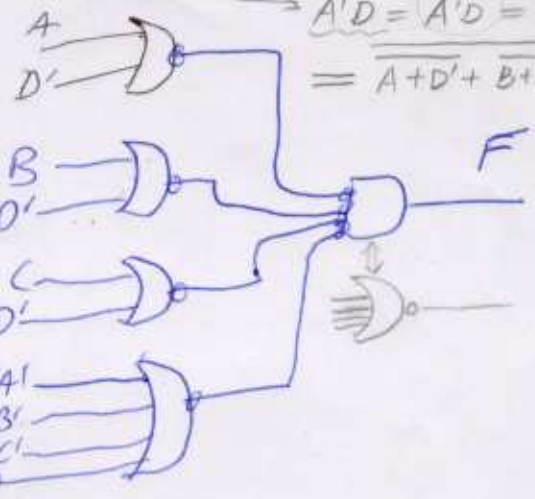
M1 $F = (\overline{A'D}) \cdot (\overline{B'D}) \cdot (\overline{C'D}) \cdot (\overline{ABCD'})$

$= (A+D')(B+D')(C+D')(A'+B'+C'+D)$

M2 $F = \overline{A'D + B'D + C'D + ABCD'}$

$A'D = \overline{\overline{A'D}} = \overline{A+D'}$

$= \overline{A+D' + B+D' + C+D' + A'+B'+C'}$

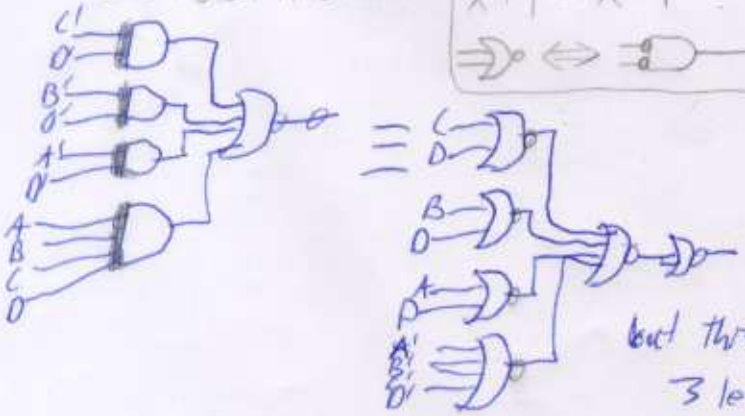


If they Circle 1's (Sum of products)

$F = C'D' + B'D' + A'D' + ABCD$

$= \overline{C+D} + \overline{B+D} + \overline{A+D} + \overline{A+B+C+D}$

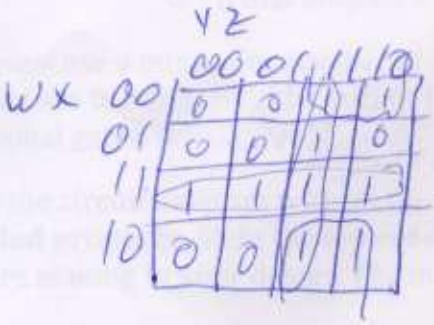
$\overline{X+Y} = \overline{X} \cdot \overline{Y}$



but this is 3 levels and uses 6 gates.

3. Simplify the following Boolean function using a Karnaugh map: (4 marks)

F(w,x,y,z) = xyz + wy + wxy' + x'y



$F = wx + yz + x'y$

4. Using 2x4 decoders and one 4x1 multiplexer, design a circuit with the following properties: It has two 2-bit binary number inputs (x_1 x_0)(y_1 y_0) and

2 inputs, ~~select~~ F_1, F_0 to

- ~~It has four modes~~ to select the function of the circuit:
  - EQ outputs a 1 if the two inputs are equal
  - GT outputs a 1 if the X is larger than Y
  - LTE outputs a 1 if the X is smaller or equal to Y
  - NULL outputs a 0

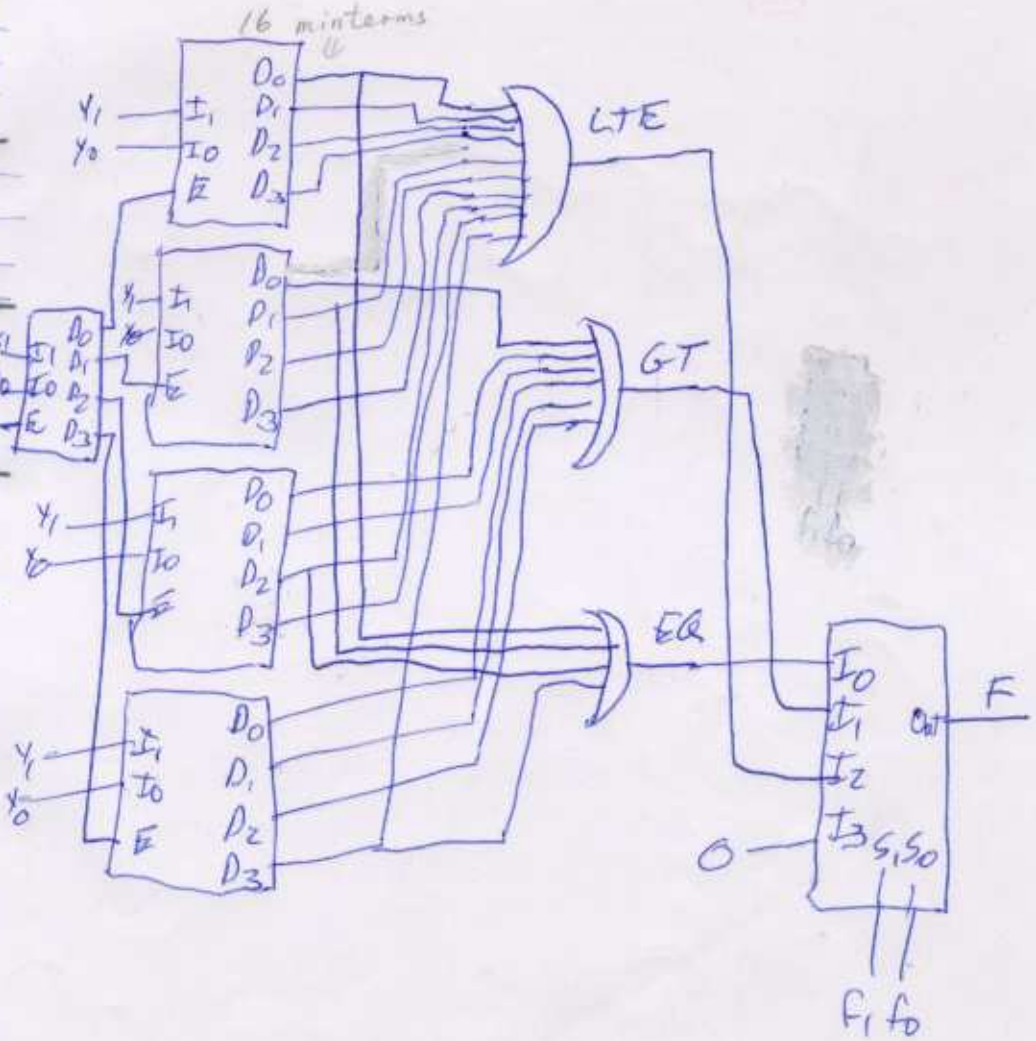
You must use a minimum number of 2x4 decoders with enables (i.e. if enable is 0 all outputs are 0, otherwise, the output is 1 for the minterm) and a minimum number of additional gates. ~~Use~~ and only one 4x1 mux.

Show the circuit diagram with all the **pins to the decoders and multiplexor labelled properly**. State ~~the type of decoders you are using and~~ any assumptions you are making in your design. (12 marks)

x_1 x_0	y_1 y_0	EQ	GT	LTE
0 0	0 0	1	0	1
0 0	0 1	0	0	1
0 0	1 0	0	0	1
0 0	1 1	0	0	1
0 1	0 0	0	1	0
0 1	0 1	1	0	1
0 1	1 0	0	0	1
0 1	1 1	0	0	1
1 0	0 0	0	1	0
1 0	0 1	0	1	0
1 0	1 0	1	0	1
1 0	1 1	0	0	1
1 1	0 0	0	1	0
1 1	0 1	0	1	0
1 1	1 0	0	1	0
1 1	1 1	1	0	1

Need a 4x16 decoder, so make from 2x4 decoders

16 minterms



function

F_1 F_0	F
0 0	EQ
0 1	GT
1 0	LTE
1 1	NULL

Extra space if needed.